**for j ← 2 to lenght[A]**

Um laço de repetição de *for* aonde se é definido a variável **j** como o valor inicial **2** que se repete até que **j** alcance o tamanho do array ordenado **A**.

**T1: j = 2 (Atribuição de valores a variáveis)  
T2: to lenght[A] (Operação lógica)**

**do key ← A[j]**

A variável **key** recebe o valor do elemento na posição **j** do array **A**. Assim. **key** armazena o valor do elemento que está sendo inserido na parte em que se corresponde o seu valor no array ordenado.

**T3: A[j] (Operação de acesso)  
T4: key ← A[j] (Atribuição de valores a variáveis)**

**▷ Insert A[j] into the sorted sequence A[1 . . j - 1]:**

Esta linha é apenas um comentário que indica que o próximo passo é inserir o valor **A[j]** na sequência ordenada que vai de **A[1]** até **A[j - 1].**

**i ← j – 1:**

A variável **i** recebe o valor da variável **j – 1**. Isso define **i** como o índice do último elemento do array.

**T5: i ← j – 1 (Atribuição de valores a variáveis)**

**while i > 0 and A[i] > key**

Um laço de repetição **while** que verifica se o valor da variável **i** é maior que zero e se o valor no Array **A** com a posição com index da variável **i** é maior que o valor na variável **key**, o último elemento do array. Então verificando se o valor na posição **i** deve ser movido para uma posição à frente para abrir espaço para key.

**T6: i > 0 (Operação lógica)  
T7: A[i] (Operação de acesso)  
T8: A[i] > key (Atribuição de valores a variáveis)**

**do A[i + 1] ← A[i]**

Caso as duas condições sejam verdade, o valor no array A com a posição de **i + 1** se torna o valor da posição de apenas **i**, deslocando o valor de **A[i]** para a posição **i + 1**, criando espaço para o **key**.

**T9: A[i] (Operação de acesso)  
T10: A[i + 1] ← A[i] (Atribuição de valores a variáveis)**

i ← i – 1

A variável **i** é decrementada em 1, passando para o próximo elemento à esquerda, para continuar a movimentação dos elementos maiores, se necessário.

**T11: i ← i – 1 (Atribuição de valores a variáveis)**

**A[i + 1] ← key**

Quando as condições do while não são mais verdadeiras, a variável **key** é inserida na posição **i + 1** do array. Assim, **key** é colocado na posição correta dentro do array ordenado

**T12: A[i + 1] ← key (Atribuição de valores a variáveis)**

O algoritmo de ordenação por inserção percorre o array, movendo cada elemento da parte não ordenada para sua posição correta na parte ordenada. Ele utiliza o laço while para deslocar os elementos maiores e, finalmente, insere o key na posição correta.

A quantidade de T (Tempos) pode aumentar caso os laços de repetição sejam executados mais de uma vez, porém, a quantidade mínima possível de tempos, sem repetições nos laços, é de 12 T´s.